

Lumiq.ai

PREPARED BY
Nikita Khoj
(2018BTechCSE028)

FACULTY GUIDE
Dr. Amit Sinhal



Department of Computer Science Engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University Jaipur

May 2022

Lumiq.ai

Practice School-2

Submitted in partial fulfilment of the requirements

for the degree of

Bachelor of Technology in Computer Science Engineering

By:

Nikita Khoj
(2018BTechCSE028)

Faculty Guide:

Dr. Amit Sinhal



Department of Computer Science Engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University Jaipur

May 2022

CERTIFICATE

This is to certify that the Practice School-2 project work entitled “**Title of the report**” submitted by **Nikita Khoj (2018BTechCSE028)**, towards the partial fulfilment of the requirements for the degree of **Bachelor of Technology in Computer Science Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for Practice School-II examination.



Dr. Amit Sinhal

Professor

Department of Computer Science
Engineering

Institute of Engineering & Technology (IET)
JK Lakshmipat University Jaipur



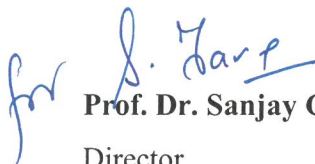
Dr. Kshitiz Verma

Associate Professor & Co-ordinator

Department of Computer Science
Engineering

Institute of Engineering & Technology (IET)
JK Lakshmipat University Jaipur

Date of Submission: 27th May 2022



Prof. Dr. Sanjay Goel

Director

Institute of Engineering & Technology
JK Lakshmipat University Jaipur

ACKNOWLEDGEMENTS

With great pleasure, I would like to extend my sincere gratitude and thanks to my parents and my sister for their sincere support they have given me.

My internship at **Lumiq.ai**, Noida was a great platform for learning, professional and personal development. I am greatly indebted to Lumiq.ai for their supervision and guidance and cooperation in project and completing it on time.

I would like to express my heartfelt gratitude to **Prof Dr. Sanjay Goel**, Professor, Department of Computer Science & Engineering and to my mentor **Dr. Amit Sinhal** for their support towards the project.

I would like to thank my industry mentor Mr. Ankur S. Chauhan for giving me opportunity to be part of the project and their consistent encouragement and assistance, Mr. Sumit, and entire team for their valuable input to the project work and making.

I would also like to thank sincerely to **Dr. Milind Thomas Themalil**, Associate Professor, Institute of Engineering and Technology and **Shounak Biswas**, Asst Director Development, JK Lakshmipat University for this internship opportunity and a platform to carry out the project.

Sincerely yours,

Nikita Khoj
(2018BTechCSE028)

ABSTRACT

The internship period consists of the bootcamp session where 15 sessions held, and assignments were given accordingly based on them. After it, I was part of project based on data analysis and reducing data point without loss of information, manual data augmentation and KPI (Key Performance Indicator) analysis.

Second project Empower, a data platform and end-to-end management product performing all the task including data extraction from data points through pipeline, data transformation and loading it into desired format. The data observability tool provides a single view of anything and everything that happens to data on the platform. As a result, it becomes a single source of truth - a reliable picture of what is happening to the information on the data platform.

LIST OF FIGURES

Figure	Page No.
Fig 2.1. SQL Architecture	8
Fig 2.2. Process with and Without a Data Catalog	10
Fig 2.3. Open Metadata Architecture	12
Fig 2.4. Open Metadata Dashboard	12
Fig 2.5. Open Metadata Components	13
Fig 2.6. Airflow Dashboard	14
Fig 2.7. Docker Architecture	16
Fig 2.8. Kafka Architecture	18

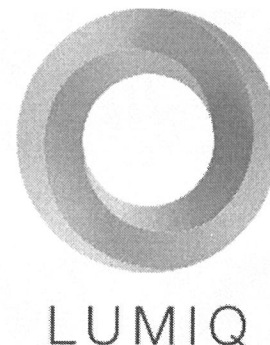
CONTENT

Certificate.....	iii
Acknowledge.....	iv
Abstract.....	v
List of Figure.....	vi
 1. Introduction	
1.1. Vision and Mission.....	1
1.2. Core vows.....	1
2. Internship Period	
2.1. Training Sessions.....	3
2.2. Key Learnings.....	4
2.3. Project 1	
2.3.1. Problem Statement.....	7
2.3.2. Methodology Adopted.....	7
2.3.3. Technology Used.....	8
2.4. Project 2	
2.4.1. Objective.....	10
2.4.2 About the Project.....	11
2.4.3 Tools and Technology Used.....	11
2.4.3.1 Open Metadata.....	11
2.4.3.2 Airflow.....	14
2.4.3.3 Docker.....	15
2.4.3.4 PostgreSQL.....	16
2.4.3.5 Kafka.....	17
2.4.3.6 Maven.....	18
3. Conclusion.....	20
Reference.....	21

Chapter 1: INTRODUCTION

1.1. Vision and Mission

- As a **data analytics company**, vision is to leverage data responsibly to build a safer, smarter, and more efficient world.
- So, being a **tech company**, the objective is a human one: to enhance the quality of human life.



- Deep business and technology understanding
- AI-powered cloud-native products
- Flexible working conditions
- Data science, data engineering, and operation
- Offices across India- Noida, Mumbai, Chennai, Pune

The vision is accomplished by keeping a hawk's eye on our mission: helping our customers succeed. Our customers are the ones who are creating and delivering the products that enhance the quality of people's lives. By helping them to bridge business needs and human purpose with highly intelligent and optimally executed technology solutions, we fulfill our vision.

1.2. Core vows

- **Integrity:** We say what we mean, and we do what we say. It's all black and white with no shades of grey. Our word is precious, and we do not give it lightly.

- **Commitment:** Everything that is entrusted to us will be done with passion and taken to its conclusion. We are self-driven, passionate, and diligent which means whatever we start, we get it done.
- **Trust:** Trust begets trust, and we intend to surround ourselves with those who trust us and who we trust. Business relationships are fruitful and satisfying when there is mutual trust on both sides.
- **Humility:** There is much to learn, and we are humbly committed to a spirit of constant learning and improvement. We will never know it all and each project, each client is a gateway to new knowledge.

Website: <https://lumiq.ai/>

Headquarters: Noida, UP

Chapter 2: INTERNSHIP PERIOD

The internship period consists of a training session which was of 2.5 months including assignments based on different topics.

2.1. Training Sessions

The training session was held on the topics:

- Internet
- Linux
- Diagram, Flowcharts, HLD, LLD
- Java
- Python
- HTML and CSS
- Time Management
- JavaScript
- GIT
- Database
- SQL
- API
- BI
- Cloud
- Docker
- Presentation and Documentation
- Spark

2.2. Key Learnings

2.2.1. Internet

- Installing a VM with OS Ubuntu using VirtualBox
- Hosting web pages using Nginx

2.2.2. Linux

- Various folders and their type of files
- Various commands of BASH language (file content command, file ownership, file permission, setting permission, data manipulation, networking command)

2.2.3. Diagram, Flowcharts, HLD, LLD

- Getting high level view of a problem
- Understanding what and why before how
- Correlate problem before solution

2.2.4. Java, Python

- Classes, Objects
- Functions.
- Constructors, Inheritance, Overloading and Overriding
- Working with regular expressions.

2.2.5. HTML and CSS

- Proper use of tags and elements
- Specificity on CSS and styling

2.2.6. JavaScript

- DOM structure
- Creating objects, storing values, and accessing in HTML
- Web Scraping
- Fetching and Creating API

2.2.7. GIT

- Basic commands (push and add files, commits, revert)
- Creating new branches and Merging branches.
- Creating conflicts and solving them.

2.2.8. Database

- SQL, NoSQL
- Graph, Time Series Databases
- Database Storages

2.2.9. SQL

- Basic Queries (Select, Insert, delete, where, having, as, in, order by, group by and nested Queries.)
- Types of Joins.
- Rank, Dense Rank

2.2.10. API

- What are APIs
- POST, GET, PUT, DELETE
- Use of APIs: Google Maps, weather apps, login authentications, payment gateways.

2.2.11. BI (Business Intelligence)

- How BI works and its process
- Identifying KPI- Key Performance Indicator
- Dashboard in Power BI
- Modelling of data in it

2.2.12. Docker

- Components of docker
- Its Functioning vs virtual machine
- Installing it in ubuntu and creating an image

2.2.13. Cloud

- AWS lambda function
- Simple Notification Service
- DynamoDB, API gateway
- Sending a message from AWS API gateway to AWS lambda

2.2.14. Presentation and Documentation

- Components of presentation and documentations
- Different types of documentations
- Communication and collaboration

2.2.15. Apache Spark

- Use Spark DataFrame API
- Writing query using SparkSQL
- Learning Spark Core

2.3. Project 1

2.3.1. Problem Statement

Data collection from huge repository and optimizing it. Performing manual data augmentation, Report and KPI analysis.

2.3.2. Methodology Adopted

- Understand the report content, data points and query and gathering them to get a systematic and homogeneous form of data for further use and optimization of report and better decision making.
- Manual data augmentation, report and KPI (Key Performance Indicator) analysis.
- work based on data analysis and reducing data point without loss of information.
- The company has dataset in form of reports which is dynamic and keep on growing based on that they make decisions for their further use. Now, when the data become so huge that they face difficulty in handling it and making decisions and other use of that.
- Therefore, the company want their record to be in a uniform and systematic format for better arrangement and management of data.
- We first have to study and analyse the report format and structure and the data points existing in it and the query formation. Then extract, segregate, and arrange the query in tables for better representation of report data without missing any piece of the information.
- Discussion with functional owners for viewpoint and recommendations.

2.3.3. Technology used

1. SQL

- SQL is a standard language for storing, manipulating, and retrieving data in databases.
- SQL stands for Structured Query Language.
- SQL can execute queries against a database, retrieve data from a database, insert records in a database, update records in a database, delete records from a database, create new databases, create new tables in a database, create stored procedures in a database, create views in a database, set permissions on tables, procedures, and views.

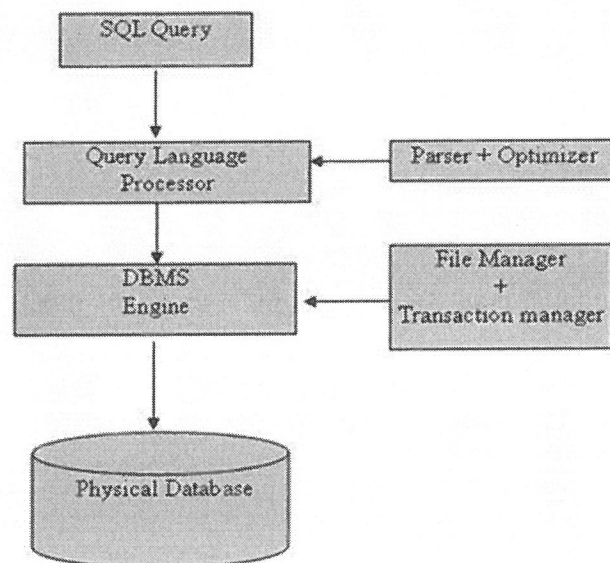


Fig 2.1. SQL Architecture

SQL Commands:

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into the following groups—

1. DDL - Data Definition Language

1. CREATE: Creates a new table, a view of a table, or other object in the database.

2. ALTER: Modifies an existing database object, such as a table.

3. DROP: Deletes an entire table, a view of a table or other objects in the database.

2. DML - Data Manipulation Language

1. SELECT: Retrieves certain records from one or more tables.

2. INSERT: Creates a record.

3. UPDATE: Modifies records.

4. DELETE: Deletes records.

3. DCL - Data Control Language

1. GRANT: Gives a privilege to user.

2. REVOKE: Takes back privileges granted from user.

2.4. Project 2

2.4.1. Objective

Implementation of data cataloging through open metadata enterprise data platform.

Metadata is data of data. Data Catalog acts as a central location of data by scanning and mapping all metadata for each data system. It is an organized inventory of data assets in the organization. It uses metadata to help organizations manage their data. It also helps data professionals collect, organize, access, and enrich metadata to support data discovery and governance.

Benefits of a Data Catalog

- Improved data efficiency
- Improved data context
- Reduced risk of error
- Improved data analysis

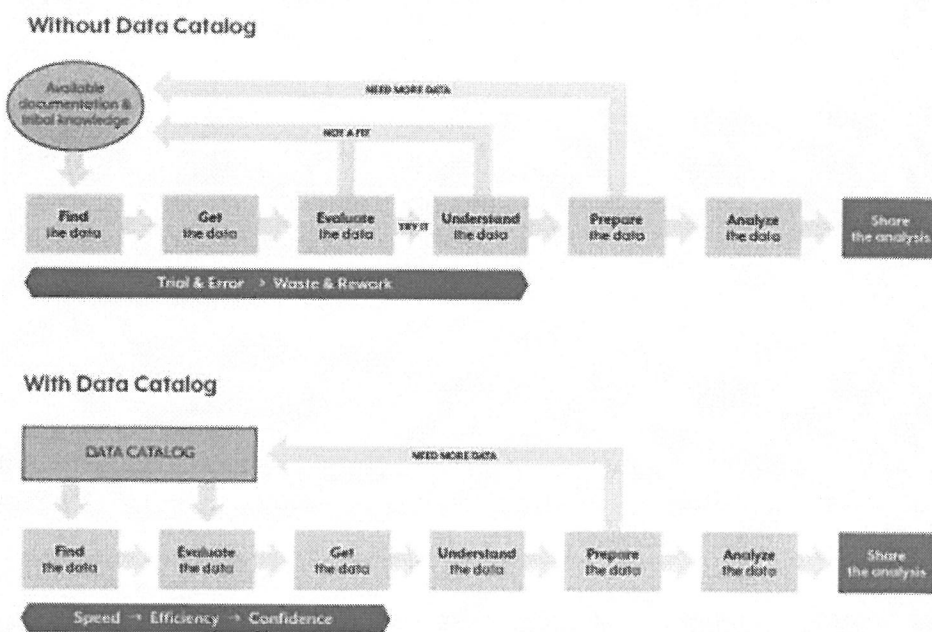


Fig 2.2: Process with and Without a Data Catalog

2.4.2 About the project

We have multiple data points coming into our data platform in almost all enterprise deployments. The data may be coming in real time or a batch mode and may be structured, semi-structured or unstructured; it may be coming in as a stream, pushed to the platform, pulled from a database, or coming via an API. These data points come through different source systems and are processed by multiple data pipelines.

The data observability tool provides a single view of anything and everything that happens to data on the platform. As a result, it becomes a single source of truth - a reliable picture of what is happening to the information on the data platform.

There are two critical components of the observability tool. The first component, the **metadata** store, contains information about all the data assets and provides a view into how the data is organized. The second part is the monitoring and exception agent, which as the name suggests, tracks, and alerts the users when an unwanted event happens.

Additionally, lineage views allow users to experience how the data is flowing and getting attached to the particular asset (for example, how the field "monthly disposable income" is calculated)

The same concept gets extended to APIs and ML models as well.

The ultimate goal of the product is not to find issues but to move in the direction of "exception driven monitoring" where users are proactively alerted about potential problems or delays that may happen. In case of incidents that have already happened, reduce the time for root cause analysis and fixing the issue.

2.4.3 Tools and Technology Used

2.4.3.1 Open Metadata

Open Metadata is an open standard with a centralized metadata store and ingestion framework supporting connectors for a wide range of services. The

metadata ingestion framework enables you to customize or add support for any service. REST APIs enable you to integrate Open Metadata with existing tool chains. Using the Open Metadata user interface (UI), data consumers can discover the right data to use in decision making and data producers can assess usage and consumer experience to plan improvements and prioritize bug fixes.

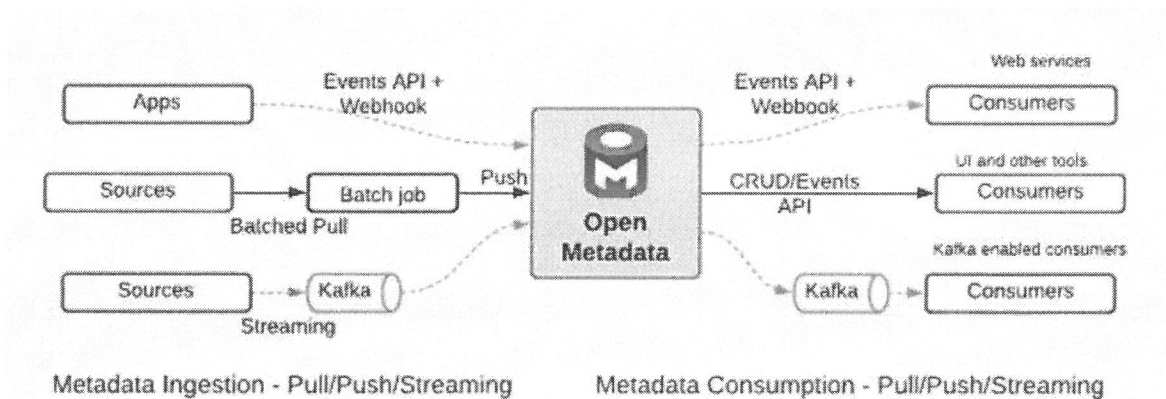


Fig 2.3. Open Metadata Architecture

Open Metadata enables metadata management end-to-end, giving you the ability to unlock the value of data assets in the common use cases of data discovery and governance, but also in emerging use cases related to data quality, observability, and people collaboration.

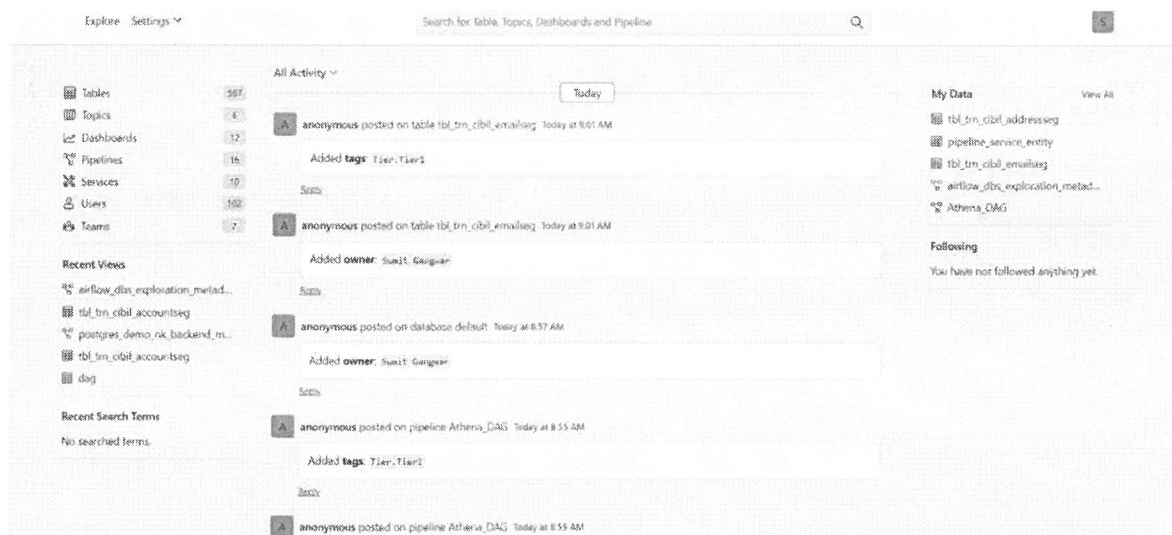


Fig 2.4. Open Metadata Dashboard

Open Metadata Components

The key components of Open Metadata include the following:

1. Open Metadata User Interface (UI) - a central place for users to discover and collaborate on all data.
2. Ingestion framework - a pluggable framework for integrating tools and ingesting metadata to the metadata store. The ingestion framework already supports well-known data warehouses.
3. Metadata APIs - for producing and consuming metadata built on schemas for User Interfaces and for Integrating tools, systems, and services.
4. Metadata store - stores a metadata graph that connects data assets and user and tool generated metadata.
5. Metadata schemas - defines core abstractions and vocabulary for metadata with schemas for Types, Entities, and Relationships between entities. This is the foundation of the Open Metadata Standard.



Fig 2.5. Open Metadata Components

2.4.3.2 Airflow

Airflow is a platform to programmatically author, schedule and monitor workflows.

Use Airflow to author workflows as Directed Acyclic Graphs (DAGs) of tasks. The Airflow scheduler executes your tasks on an array of workers while following the specified dependencies. Rich command line utilities make performing complex surgeries on DAGs a snap. The rich user interface makes it easy to visualize pipelines running in production, monitor progress, and troubleshoot issues when needed.

When workflows are defined as code, they become more maintainable, versionable, testable, and collaborative.

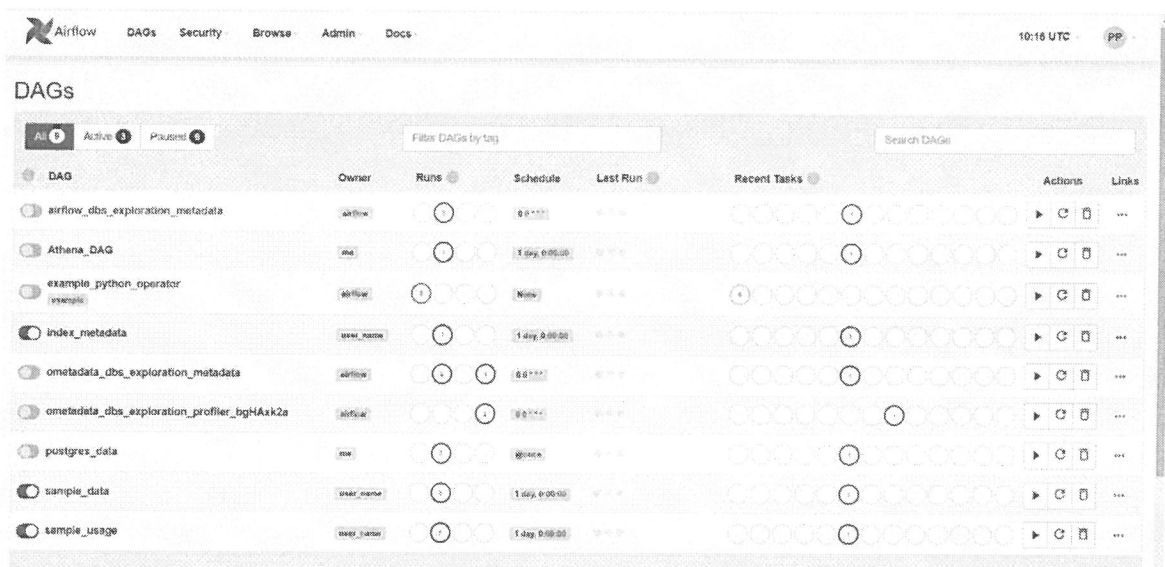


Fig 2.6. Airflow Dashboard

Principles

1. Dynamic: Airflow pipelines are configuration as code (Python), allowing for dynamic pipeline generation. This allows for writing code that instantiates pipelines dynamically.
2. Extensible: Easily define your own operators, executors and extend the library so that it fits the level of abstraction that suits your environment.

3. **Elegant:** Airflow pipelines are lean and explicit. Parameterizing your scripts is built into the core of Airflow using the powerful Jinja templating engine.

4. **Scalable:** Airflow has a modular architecture and uses a message queue to orchestrate an arbitrary number of workers. Airflow is ready to scale to infinity.

DAG

DAG abbreviates for Directed Acyclic Graph. It is the heart of the Airflow tool in Apache. It can be specifically defined as a series of tasks that you want to run as part of your workflow. The Airflow tool might include some generic tasks like extracting out data with the SQL queries or doing some integrity calculation in Python and then fetching the result to be displayed in the form of tables. In Airflow, these generic tasks are written as individual tasks in DAG. The main purpose of using Airflow is to define the relationship between the dependencies and the assigned tasks which might consist of loading data before executing. It might also consist of defining an order of running those scripts in a unified order.

2.4.3.3 Docker

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run.

Docker is an OS virtualized software platform that allows IT organizations to easily create, deploy, and run applications in Docker containers, which have all the dependencies within them. The container itself is just a very lightweight package that has all the instructions and dependencies—such as frameworks, libraries, and bins—within it.

The container itself can be moved from the environment to the environment very easily. In a DevOps life cycle, the area where Docker really shines is deployment, because when you deploy your solution, you want to be able to guarantee that the code that has been tested will actually work in the production environment. In addition to that, when you're building and testing the code, having a container running the solution at those stages is also beneficial because you can validate your work in the same environment used for production.

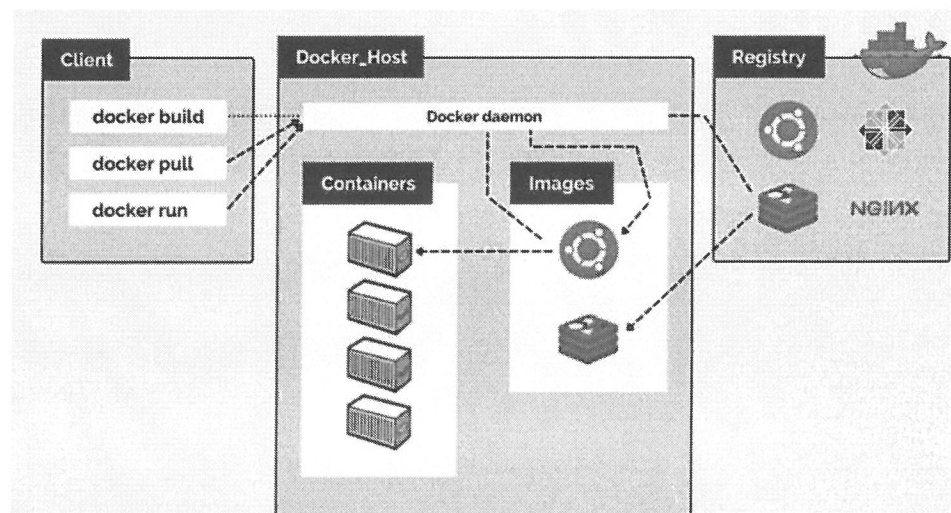


Fig 2.7. Docker Architecture

2.4.3.4 PostgreSQL

PostgreSQL is a powerful, open-source object-relational database management system (ORDBMS)

PostgreSQL is an open-source descendant of this original Berkeley code. It supports a large part of the SQL standard and offers many modern features:

- complex queries
- foreign keys
- triggers
- updatable views
- transactional integrity
- multiversion concurrency control

Also, PostgreSQL can be extended by the user in many ways, for example by adding new

- data types
- functions
- operators
- aggregate functions
- index methods
- procedural languages

PostgreSQL supports four standard procedural languages, which allows the users to write their own code in any of the languages and it can be executed by PostgreSQL database server.

2.4.3.5 Kafka

Apache Kafka is a community distributed event streaming platform capable of handling trillions of events a day. Initially conceived as a messaging queue, Kafka is based on an abstraction of a distributed commit log.

In Big Data, an enormous volume of data is used. Regarding data, we have two main challenges. The first challenge is how to collect large volume of data and the second challenge is to analyse the collected data. To overcome those challenges, you must need a messaging system.

Kafka is designed for distributed high throughput systems. Kafka tends to work very well as a replacement for a more traditional message broker. In comparison to other messaging systems, Kafka has better throughput, built-in partitioning, replication and inherent fault-tolerance, which makes it a good fit for large-scale message processing applications.

Apache Kafka is a distributed publish-subscribe messaging system and a robust queue that can handle a high volume of data and enables you to pass messages from one endpoint to another. Kafka is suitable for both offline and online message consumption. Kafka messages are persisted on the disk and replicated within the cluster to prevent

data loss. Kafka is built on top of the ZooKeeper synchronization service. It integrates very well with Apache Storm and Spark for real-time streaming data analysis.

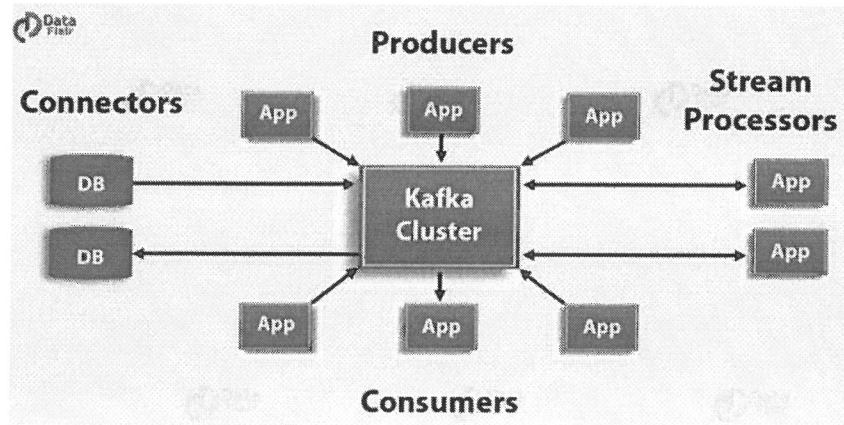


Fig 2.8. Kafka Architecture

2.4.3.6 Maven

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Maven's primary goal is to allow a developer to comprehend the complete state of a development effort in the shortest period of time. In order to attain this goal, Maven deals with several areas of concern:

- Making the build process easy
- Providing a uniform build system
- Providing quality project information
- Encouraging better development practices

Maven Objective

Maven's purpose is to provide developers with:

- A comprehensive, maintainable, reusable, and simple model for projects.
- A set of tools and plug-ins that can interact with the declarative model.

Maven is chiefly used for Java-based projects, helping to download dependencies, which refers to the libraries or JAR files. The tool helps get the right JAR files for each project as there may be different versions of separate packages.

Maven provides useful project information that is in part taken from your POM and in part generated from your project's sources. For example, Maven can provide:

- Change log created directly from source control
- Cross referenced sources
- Mailing lists managed by the project
- Dependencies used by the project
- Unit test reports including coverage

Third party code analysis products also provide Maven plugins that add their reports to the standard information given by Maven.

Chapter 3: CONCLUSION

Managing data in the age of big data, data lakes, and self-service is challenging. Data catalogs help to step up to those challenges. Active data curation is a core element of data catalog success and a critical practice for modern data management.

The ultimate goal of the product is not to find issues but to move in the direction of "exception driven monitoring" where users are proactively alerted about potential problems or delays that may happen. In case of incidents that have already happened, reduce the time for root cause analysis and fixing the issue.

REFERENCES

1. W3schools.com. 2022. SQL Tutorial. [online] Available at: <<https://www.w3schools.com/sql/#:~:text=SQL%20is%20a%20standard%20language,Postgres%2C%20and%20other%20database%20systems.>>>
2. Moderndatastack.xyz. 2022. *Data Cataloging - Modern Data Stack | Modern Data Stack*. [online] Available at: <<https://www.moderndatastack.xyz/category/data-cataloging>>
3. Alation. 2022. *What Is a Data Catalog? - Importance, Benefits & Features | Alation*. [online] Available at: <<https://www.alation.com/blog/what-is-a-data-catalog/>>
4. Docs.open-metadata.org. 2022. *Introduction - OpenMetadata*. [online] Available at: <<https://docs.open-metadata.org/overview/readme>>
5. Apache Airflow. 2022. *Documentation*. [online] Available at: <<https://airflow.apache.org/docs/>>
6. Docker Documentation. 2022. *Docker Documentation*. [online] Available at: <<https://docs.docker.com/>>
7. PostgreSQL.org. 2022. *PostgreSQL: Documentation*. [online] Available at: <<https://www.postgresql.org/docs/>>
8. Maven.apache.org. 2022. *Maven - Introduction*. [online] Available at: <<https://maven.apache.org/what-is-maven.html>>

INTERNSHIP CERTIFICATE

Date: 20/05/2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Nikita Khoj has done her internship in Big Data at Lumiq.ai, Noida, from 3rd Jan' 2022 to 20th May 2022 for the duration of 4.5 months.

She has worked on a project titled Empower. This project was aimed at implementation of data catalog through Open Metadata for enterprise data platform. As part of the project, she has worked on Open Metadata framework and connecting services with it.

During her internship she has demonstrated her skills with self-motivation to learn new skills. Her performance exceeded our expectations, and she was able to complete the project on time.

We wish her all the best for her upcoming career.

Lumiq.ai

Sneha Pandey

HR

[Signature] [Seal of Company]

